Appendix 2-A. Application and System Development Requirements

Introduction

AHRQ uses a System Development Lifecycle (SDLC) framework that is consistent with the HHS Enterprise Lifecycle Framework (EPLC). This framework provides the basis for conducting application development projects using sound project management principles and industry best practices. The SDLC framework provides a disciplined approach that employs the following traditional project phases:

- Initiation

- Concept

- Planning

- Requirements analysis

- Design

- Development

- Testing

- Implementation/deployment

- Operations and maintenance

- Disposition/retirement

It's important to note that no single development method is appropriate for all projects. The most appropriate approach for delivery will be heavily influenced by the unique characteristics of the performing organization and the project itself, and may not be exactly what was used to successfully deliver past projects. The HHS EPLC Framework recognizes and allows for this flexibility through the use of a Project Process Agreement document that allows for the tailoring of the EPLC Framework to meet any unique project needs and/or approaches.

Contractors are not required to follow a specific development methodology; however, the contractor's SDLC must be capable of producing AHRQ-required system deliverables containing the required content as described further in the following section. The contractor is also required to obtain Federal project officer approval before moving from one project phase to another. This approval process corresponds to stage gate reviews in the HHS EPLC model. The contractor must also conform to AHRQ Configuration Management (CM) and change control standards, which require appropriate controls for managing software and documentation baselines, changes to software artifacts using an

appropriate Integrated Development Environment or version management tool, document change requests, and obtaining approval through a formal change control process that requires Federal project officer and possible AHRQ IT approval prior to implementation.

Learn more about the HHS EPLC framework.

Table 1 identifies the documentation deliverables required for all IT projects and the content required for each deliverable.

Table 1—Documentation Deliverables

Deliverable	AHRQ Life Cycle Phase	Formats
Project Initiation Document	Project Initiation	Microsoft® Word
Project Work Plan	Project Planning	Microsoft® Project
System Requirements Document	Requirements and Analysis	Rational Requisite Pro, Microsoft® Word
Requirements Traceability Matrix	Requirements and Analysis	Rational Requisite Pro, Microsoft® Word
System Design Document	Software Design	Rational Software Modeler, Microsoft® Word, Rational Software Architect, and so forth
Test Plan	Testing	Microsoft® Word
Test Scripts	Testing	Microsoft® Word, Rational Test Manager, and so forth
User Acceptance Testing	Report Implementation	Microsoft® Word
User Guide	Deployment	Microsoft® Word
Operations Manual	Deployment	Microsoft® Word
Version Description Document	Deployment	Microsoft® Word

System Documentation

The contractor will provide system documentation to the Agency for all proposed hardware, software, security, backup/recovery, and other IT infrastructure and components and solutions needed to support a project. The documentation is to be delivered to the Federal project officer for review and approval for each release. All documentation will be baselined with each system release. In addition, the source code for each production release will be delivered and stored in the same project library as the documentation artifacts. The contractor will be required to update these baselined

artifacts for each production release of the system. Sample documents and templates for the required documentation artifacts are available as guidance to the contractor. The following documents as mentioned in Table 1, "Documentation Deliverables," are required by AHRQ.

Project Initiation Document

The Project Initiation Document (PID) is intended to be a statement of purpose and scope for initiating a given project and a guide to manage expectations in both process and deliverables throughout the SDLC. The PID defines the business case for the project by defining the purpose; milestones; resources; objectives; costs; risks, including mitigation strategies; and the artifacts and IT technologies (architecture) utilized and produced for, and during, the project. The PID serves as the formal funding commitment document approved by the Contracting Officer's Representative (COR) and stakeholders. Additionally, the PID must be approved by AHRQ IT management, and in some cases, the AHRQ Information Technology Review Board for technical viability; adherence to Agency Enterprise Architecture; technical standards; and formal project management requirements as derived from departmental standards and accepted Project Management Institute Project Management Body of Knowledge standards. In the case of external development contracts, the PID can be satisfied by the formal proposal submitted by the vendor and accepted by AHRQ.

Project Work Plan

The System Project Plan or Project Work Plan (PWP) provides a method to assign and track the project resources, hours, and specific deliverables. This plan provides the detailed Work Breakdown Structure and resource loading that can be used to identify project costs and is intended for the project manager to track the schedule and cost of a project, including development of earned value management measures. The PWP is delineated by the phases of the project that include project initiation, generation of the PWP schedule, requirements gathering, system design, system development, system testing, including user acceptance, system deployment and system support, and production of project deliverables that require COR or stakeholder acceptance and signoff to continue project tasks identified in the PWP.

System Requirements Document

The System Requirements Document (SRD) contains the system requirements, use cases, and supplementary specifications that provide the basis for design and development of the system. The following information is provided for each requirement identified in the document:

- Requirement ID, name, and title

- Requirement description

- Software release version

- Use case model

- Use case specifications

- Supplementary specifications

A text-based functional requirements document may be provided instead of a use case model, use case specifications, and supplementary specifications.

Requirements Traceability Matrix

The requirements traceability matrix associates requirements with the work products that satisfy them. This matrix is created at the beginning of a project's lifecycle to trace the requirements from identification through testing. The project elements are traced as they relate to other project elements, especially those related to requirements.

The purpose of establishing traceability is to help understand the sources of requirements, manage the scope of the project, manage changes to requirements, assess the project impact of a change in a requirement, and verify that all requirements of the system are fulfilled by the implementation.

The following values are required for the traceability matrix:

- Requirement ID and title

- Version of the system in which the requirement will be implemented

- Use case to which the requirement can be traced

- Version of the design document in which the requirement is implemented

- ID of the test script in which the requirement is tested

- Version number of the source code in which the requirement is implemented

Figure 1 shows a sample of the data traced through a project's lifecycle.

Figure 1—Data Traced Through Project Lifecycle

Requirements:	Version	Trace To UC	Trace to Design	Trace to Test	Trace to Source	CR	Status
▶ FEAT8: The system shall display the Principal... The system shall capture and display the Principal Investigator's name on the Quarterly Report.	2.00.00	UC7, UC13				Prod00000098,Prod00000	Incorporate
FEAT9: The system shall display Principal... The system shall capture and display the Principal Investigator's Address.	2.00.00	UC7, UC13				Prod00000098,Prod00000	Incorporate
FEAT10: The system shall display Principal... The system shall display and capture the Principal Investigator's telephone number.	2.00.00	UC7, UC13				Prod00000262	Incorporate
FEAT11: The system shall display Principal... The system shall capture and display the E-mail address of the Principal Investigator.	2.00.00	UC7, UC13				Prod00000262	Incorporate
FEAT12: The system shall display the Principal... The system shall capture and display the main fax number for the Principal Investigator.	2.00.00	UC7, UC13				Prod00000262	Incorporate
⊟ FEAT13: the system shall display and Track... The system shall capture and track milestones for a given project/grant. HIT uses the word Milestone while PS uses...	2.00.00	UC11, UC13				Prod00000268	Proposed
FEAT13.1: The system shall display and track... The system shall capture and track overall progress of project milestones and shall display these in the report.	2.00.00	UC11				Prod00000268	Proposed
FEAT13.2: The system shall display and track... The system shall capture and display milestone barriers.	2.00.00	UC11				Prod00000272	Proposed

System Design Document

The system design document (SDD) details the design and implementation of all custom software features of the system. The design descriptions must include use cases that detail the interaction that occurs between a user and the system.

The document describes the general nature of the system and describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. For each significant package, a section of the document should detail its decomposition into classes and class utilities. Architecturally significant classes should be introduced and a description of their responsibilities should accompany the introduction. Any significant relationships, operations, and attributes should be detailed in this document.

The document should be organized by use case, so that it provides traceability back to the initial requirements. The document must also contain a description of the database model and data elements used to support the application. These data can be referenced to an appropriately maintained entity relationship diagram and data definitions which conform to AHRQ CM standards.

Test Plan

The test plan defines the approach for testing a particular application or system. It is a high-level description of the testing process to be performed. The test plan outlines the types of testing to be performed, the requirements to be tested, the test environment, testing tools, pass/fail criteria, and a risk assessment. At a minimum, the document should contain the following:

Test Description

- General overview of the plan for testing the entire system

- Test objectives for all testing levels (e.g., module, unit)

- Scope and guiding principles for the testing effort

- Policy for resolving conflicts that arise during the testing process

Acceptance Criteria
- Criteria agreed upon with the customer for acceptance of the software

Approach
- How each major group of software features will be adequately tested

- Major testing activities, techniques, and testing tools

- Test environment—hardware, network, software, and test database

Tasks
- Individual tasks that must be performed

- Individual or organization responsible for each task

Schedule, Resources, and Milestones

Test Scripts

The test scripts define testing scenarios completed for an application. Each scenario details the steps to be performed, expected results, and pass/fail criteria. At a minimum, the document should contain the following:

- Test script identifier

- Test description

- Test objective

- Test environment/setup including any required data such as login credentials, etc.

- Mapping to specific requirements and design elements contained in the SRD and SDD

- Step sequences and actions

- Expected results

- Pass/fail criteria

- Actual results

- Comments

User Acceptance Test Report

The user acceptance testing (UAT) report should include a summary of the testing environment (hardware, software, tools, participant list, etc.) and procedures used to demonstrate and obtain stakeholder approval of the application or system prior to production deployment. The UAT report should contain a mapping to the SRD and SDD items included in the release as well as an exception list or identified change requests that were generated as the result of testing.

User Guide

The user guide should be completed prior to production. The user guide is a "how to" manual that navigates the user in detail through the use of the application. This document usually contains system screen shots and provides step-by-step instructions for completing tasks and activities. It is written on a business level with the needs of the user in mind. At a minimum, the document should contain the following content:

- Introduction

- Summary of the application

- Glossary (definitions/acronyms)

- Procedures (step-by-step instructions on how to use the system)

- Troubleshooting tips

Operations Manual

The operations manual provides guidance and defines procedures related to the operational implementation of the system. At a minimum, the document should contain the following:

- System overview

- Statement of acceptable use of the system and information

- Hardware and software descriptions

- Interfaces with other systems and databases

- Access and authentication requirements

- System configuration and administration procedures

- Security procedures, including virus protection

- Incident reporting and handling

- System startup and recovery procedures

- Change management procedures

Version Description Document

The version description document identifies and describes the general release information and inventory of software released (bill of materials), for a specific application, including prototype iterations. The document should include the following sections listed below:

- Introduction: Describes the objective of the document, defines the release identification, and provides contact information

- General release information: Provides information about the specific release, including any interfaces and dependencies

- Installation instructions: Describes the steps required to install the software

- Version description: Provides an inventory of List Objects and Module Types (e.g., class files, SQL Scripts, HTML files, DTD, and XML files)

- Recovery instructions: Describes the steps required to reconstruct the release from the product baselines, established in the configuration management library